this binding

Q1

```
"use strict";
function foo() {
  console.log(this);
}
foo();
```

Q2

```
function foo() {
  console.log(this.a);
}
var a = 10;
foo();
```

Q3

```
"use strict";
var a = 20;
function test() {
  console.log(this.a);
}
test();
```

Q4

```
const obj = {
  a: 5,
  show() {
    console.log(this.a);
  }
};
```

```

obj.show();
```

Q5

```
const user = {
  name: "John",
  show() {
    console.log(this.name);
  }
};

const fn = user.show;
fn();
```

Q6

```
let obj = {
  x: 100,
  inner: {
    x: 200,
    print() {
      console.log(this.x);
    }
  }
};

const p = obj.inner.print;
p();
```

Q7

```
function foo() {
  console.log(this.x);
}
const a = { x: 1 };
const b = { x: 2 };

foo.call(a);
foo.call(b);
foo();
```

Q8

```
function sum() {
  return this.a + this.b;
}
const obj = { a: 10, b: 20 };
const f = sum.bind(obj);

console.log(f());
console.log(sum());
```

Q9

```
const user = {
  age: 25,
  print() {
    console.log(this.age);
  }
};

user.print.call({ age: 99 });
```

Q10

```
function foo() {
  console.log(this.v);
}

const a = foo.bind({ v: 1 });
const b = a.bind({ v: 2 });

b();
```

Q11

```
const obj = {
  x: 10,
  show: () => {
    console.log(this.x);
  }
};
obj.show();
```

Q12

```
const obj = {
  x: 10,
  show() {
    const inner = () => console.log(this.x);
    inner();
  }
};

obj.show();
```

Q13

```
function Person() {
  this.age = 20;
  setTimeout(function () {
    console.log(this.age);
  }, 0);
}

new Person();
```

Q14

```
function Person() {
  this.age = 20;
  setTimeout(() => {
    console.log(this.age);
  }, 0);
}

new Person();
```

Q15

```
var x = 100;
const obj = { x: 200 };

const foo = () => console.log(this.x);

foo.call(obj);
```

Q16

```
function User() {
  this.name = "Alice";
}
const u = User();
console.log(u);
console.log(name);
```

Q17

```
class A {
  constructor() {
    this.x = 10;
  }
  show() {
    console.log(this.x);
  }
}

const f = new A().show;
f();
```

Q18

```
class Test {
  x = 10;

  static x = 20;

  show() {
    console.log(this.x);
  }

  static show() {
    console.log(this.x);
  }
}

const t = new Test();
t.show();
```

```
Test.show();
```

Q19

```
const obj = {
  count: 0,
  inc() {
    setTimeout(function () {
      console.log(this.count);
    }, 0);
  }
};
obj.inc();
```

Q20

```
const obj = {
  count: 0,
  inc() {
    setTimeout(function () {
      console.log(++this.count);
    }.bind(this), 0);
  }
};

obj.inc();
```

Q21

```
const obj = {
  arr: [1, 2, 3],
  sum() {
    return this.arr.map(function (v) {
      return v + this.inc;
```

☐ }, { inc: 5 });

```
  }
};

console.log(obj.sum());
```

Q22

```
function A() {
  this.v = 10;
}
A.prototype.show = function () {
  console.log(this.v);
};

const a = new A();
const method = a.show;

method();
```

Q23

```
function B() {
  this.v = 77;
}
B.prototype.show = () => {
  console.log(this.v);
};

new B().show();
```

Q24

```
var length = 10;
```

```
function fn() {
  console.log(this.length);
}

const obj = {
  length: 5,
  method(fn) {
    fn();
  }
};

obj.method(fn);
```

Q25

```
var length = 4;

function fn() {
  console.log(this.length);
}

const o = {
  length: 5,
  method(...args) {
    args[0]();
  }
};

o.method(fn, 1, 2);
```

Q26

```
function test() {
  return {
    name: "A",
    print: function () {
      console.log(this.name);
    }
  };
}

const a = test();
const p = a.print;
p();
```

Q27

```
const obj = {
  x: 10,
  getX() {
    return this.x;
  }
};

const y = {
  x: 50,
  getX: obj.getX
};

const fn = y.getX;
console.log(fn());
```

Q28

```
const foo = () => console.log(this.val);
const bar = foo.bind({ val: 10 });
```

```
bar();
```

Q29

```
var a = 1;

const obj = {
  a: 2,
  f: function () {
    return () => {
      console.log(this.a);
    };
  }
};

const fn = obj.f();
fn();
```

Q30

```
var a = 1;

const obj = {
  a: 2,
  f: function () {
    return function () {
      console.log(this.a);
    };
  }
};

const fn = obj.f();
```

```
fn();
```

Q31

```
let a = 10;
eval("a = 20");
console.log(a);
```

Q32

```
"use strict";
let a = 10;
eval("a = 30");
console.log(a);
```

Q33

```
var x = 5;
function test() {
  eval("var x = 100");
}
test();
console.log(x);
```

Q34

```
function foo() {
  eval("var x = 10");
  console.log(x);
}
foo();
```

Q35

```
function foo(a) {
  eval("a = a + 10");
  console.log(a);
}
foo(5);
```